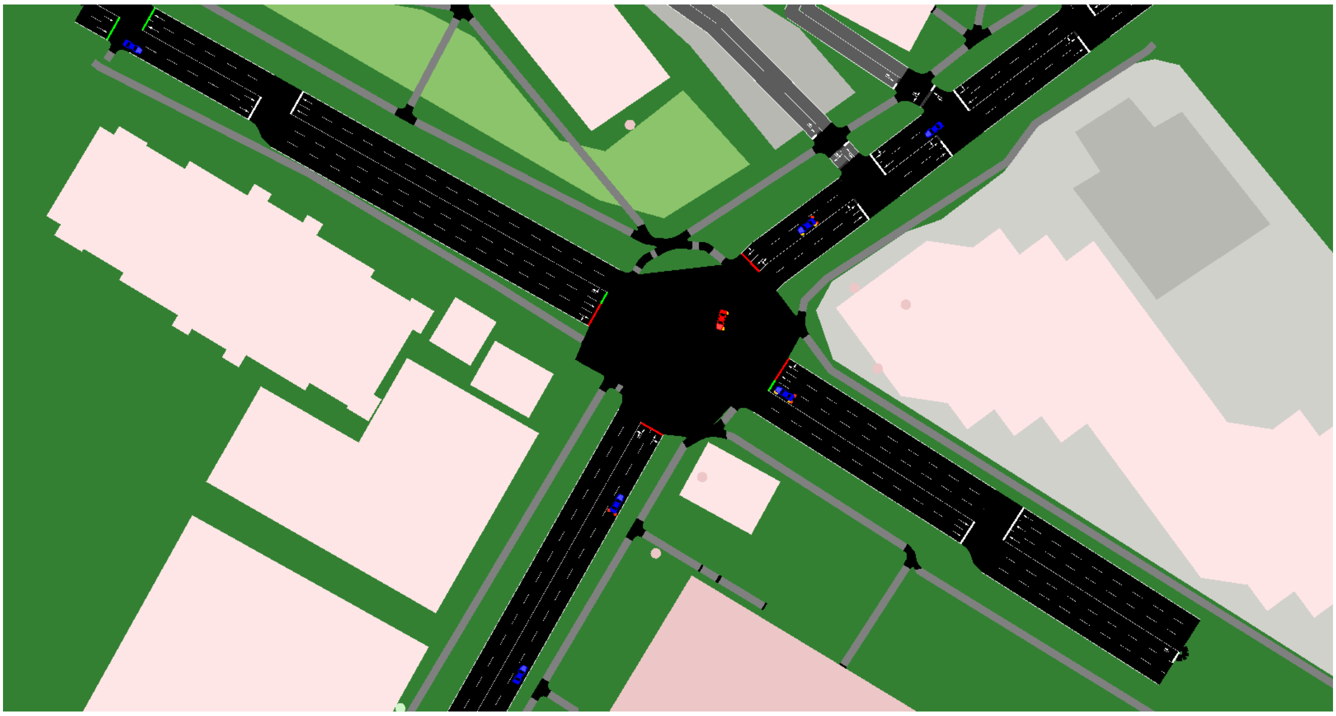
Weekly Updates (14/12/23) – Vinayak Gajendra Panchal

Ego vehicle collision prediction – Statistical Model Approach (VECM)

Simulation setup:

Step-length	0.10 sec (10 cycles/sec)
Collision.action	remove
Collision.stoptime	2 sec
Collision.check-junctions	True
'- - log' (simulation log files)	simulation_main.log
Number of Vehicles	11 (1 ego-vehicle, 10 non-ego vehicle)

4-Way Junction Details:



The SUMO (Simulation of Urban MObility) map conversion web interface, which converts OpenStreetMap (OSM) to SUMO network files, is used to create a simulation of a 4-way intersection from the Kitchener-Waterloo region, specifically at the Weber Street East - Frederick Street intersection. This tool extracts essential elements of real-world road networks within a defined geographical area, including traffic lights, pedestrian pathways, and road lanes, accurately reflecting the actual map layout.

Approach: Our approach is to forecast the upcoming x and y positions of each vehicle, both the ego and non-ego vehicles, for a duration of 2 seconds (equivalent to 20 steps). This prediction will be based on the vehicle's preceding x and y coordinates observed over a period of 4 seconds (42 steps). This forecasting will be done using the Vector Error Correction Model (VECM) which is a cointegrated Vector Auto Regressive model (VAR), better for non-stationary time series data. The prediction data, which forecasts vehicle positions for 20 steps, will be aggregated to obtain average x and y coordinates for each vehicle, approximately every second. This results in about 2 to 3 seconds of averaged x and y positions per vehicle. Utilizing these aggregated positions, the relative distances (Euclidian Displacement) between each vehicle and the ego vehicle will be calculated. A distance of less than 1 meter between any vehicle and the ego vehicle will be interpreted as an indication of a collision.

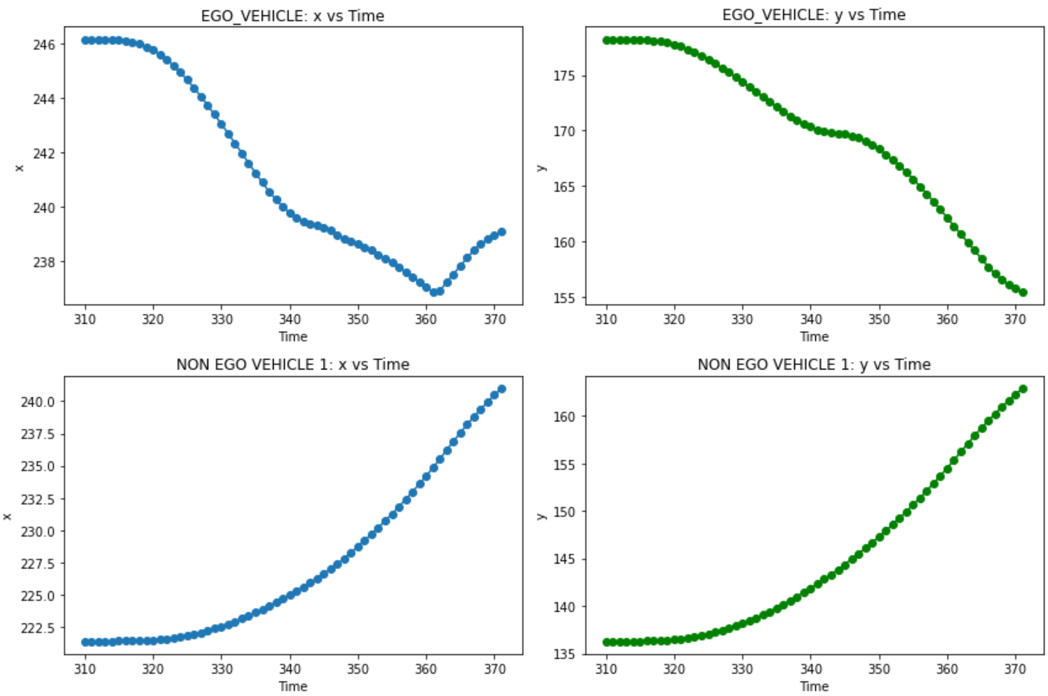
Dataset Creation: To predict the next x and y coordinates of the vehicles in the simulation we first need to extract all the x and y coordinates of the corresponding vehicles at each timestamp from the FCD data.

	time	non_ego_flow.0_x	non_ego_flow.0_y	non_ego_flow.1_x	non_ego_flow.1_y	non_ego_flow.2_x	non_ego_flow.2_y	non_ego_flow.3_x	non_ego_flow.3_y
0	0.0	352.17	91.54	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0
1	0.1	352.17	91.54	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0
2	0.2	352.16	91.54	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0
3	0.3	352.15	91.54	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0
4	0.4	352.14	91.55	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0	-1000.0

Dataset features:

1. **Time:** Time in seconds when these vehicles were present in the simulation.
2. **ego_x, ego_y:** spatial data of ego-vehicle at all timestamps.
3. **non_ego_flow.X_x, non_ego_flow.X_y:** spatial data of non-ego vehicle at all timestamps. Here X represents whole numbers representing vehicle number.
4. If any vehicle is not present in that timestamp, a value of -1000 to both x and y were given.

The dataset was divided into chunks of size 62 rows, where 42 will be used as lag data to predict the 20 next x and y values of the vehicle present at that timeframe. In these chunks only vehicles present in that timestamp will be considered i.e. vehicle x and y columns with more than 10 instances of -1000 will be dropped.



Analysis: To check whether the approach works correctly, I chose the data chuck (62 rows) where the actual collision happened between the ego_vehicle and non-ego vehicle with a relative distance of 0.9 as calculated from FCD data. Here we will check whether the VECM model can be used to predict x and y which lead to collision.

1. **Cointegration test:** Johansen test for finding cointegrating relationships between two or more time-series data. When the number of lagged differences in the model was kept at 5 (k_ar_diff = 5) provided the best results as shown below. For this, we split the data into 4 parts vertically each with 5 distinct columns.

```
Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
non_ego_flow.1_x ::  412.68    > 60.0627   =>   True
non_ego_flow.1_y ::  259.64    > 40.1749   =>   True
non_ego_flow.2_x ::  131.12    > 24.2761   =>   True
non_ego_flow.2_y ::  61.44     > 12.3212   =>   True
non_ego_flow.3_x ::  5.49      > 4.1296    =>   True
Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
non_ego_flow.3_y ::  262.55    > 60.0627   =>   True
non_ego_flow.4_x ::  144.54    > 40.1749   =>   True
non_ego_flow.4_y ::  60.01     > 24.2761   =>   True
non_ego_flow.5_x ::  22.18     > 12.3212   =>   True
non_ego_flow.5_y ::  8.08      > 4.1296    =>   True
Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
non_ego_flow.6_x ::  432.22    > 60.0627   =>   True
non_ego_flow.6_y ::  171.85    > 40.1749   =>   True
non_ego_flow.7_x ::  71.53     > 24.2761   =>   True
non_ego_flow.7_y ::  37.57     > 12.3212   =>   True
non_ego_flow.8_x ::  11.17     > 4.1296    =>   True
Name   ::  Test Stat > C(95%)    =>   Signif
 ----------------------------------------
non_ego_flow.8_y ::  268.32    > 60.0627   =>   True
non_ego_flow.9_x ::  135.46    > 40.1749   =>   True
non_ego_flow.9_y ::  73.03     > 24.2761   =>   True
ego_x  ::  29.36     > 12.3212    =>   True
ego_y  ::  1.01      > 4.1296    =>   False
```

2. **Augmented Dickey-Fuller test:** This test was used to see if the time series columns are stationary or not. On running this test, we found that in most of the columns, the time series data is non-stationary. Since the data is non-stationary, we shifted our strategy from the VAR or VARMA models to the VECM model, which can handle non-stationary time series.

Results: We applied VECM on 4 different splits and then predicted the next 20 steps of x and y of each vehicle, these steps were then concatenated into one whole prediction dataframe. We also calculated the mean square error (MSE) with the actual data.

X-coordinate	MSE	Y-coordinate	MSE
non_ego_flow.1_x	4.598198	non_ego_flow.1_y	4.940872
non_ego_flow.2_x	0.357967	non_ego_flow.2_y	0.026167
non_ego_flow.3_x	0.000725	non_ego_flow.3_y	0.004007
non_ego_flow.4_x	0.000505	non_ego_flow.4_y	0.000565

non_ego_flow.5_x	0.037494	non_ego_flow.5_y	0.147890
non_ego_flow.6_x	0.003121	non_ego_flow.6_y	0.000979
non_ego_flow.7_x	0.449876	non_ego_flow.7_y	10.204445
non_ego_flow.8_x	14.919871	non_ego_flow.8_y	23.549332
non_ego_flow.9_x	0.241920	non_ego_flow.9_y	0.108315
ego_x	0.276799	ego_y	0.295962

The Mean Squared Error (MSE) analysis indicates that the VECM (Vector Error Correction Model) was highly effective in predicting the trajectory of most vehicles, with the exception of vehicles 7 and 8. Additionally, the model demonstrated satisfactory performance in forecasting interactions between the ego vehicle and non-ego vehicle 1, referred to as conflict vehicles.

Given that the predictions are made in increments of 0.1 steps, which is not practically applicable for predicting collisions in real-world scenarios that typically don't operate on sub-second timescales, the predicted x and y values for all vehicles were averaged and rounded to the nearest whole second. This approach effectively condensed the predictions to approximately 3-second intervals, exemplified by time values like 36.0, 37.0, and 38.0 seconds, as illustrated in the data shown below.

Averaged Predicted next steps:

	Time_S	non_ego_flow.1_x	non_ego_flow.1_y	non_ego_flow.2_x	non_ego_flow.2_y	non_ego_flow.3_x	non_ego_flow.3_y	non_ego_flow.4_x
0	36.0	231.777348	151.149689	158.051618	37.187874	170.113492	58.642145	204.644729
1	37.0	238.986656	159.912464	153.296190	28.041533	164.513436	48.688127	204.625724
2	38.0	245.596184	167.830092	150.385540	21.951763	160.839773	42.072405	204.609227

Actual Next steps:

	Time_S	non_ego_flow.1_x	non_ego_flow.1_y	non_ego_flow.2_x	non_ego_flow.2_y	non_ego_flow.3_x	non_ego_flow.3_y	non_ego_flow.4_x
0	36.0	231.585	151.0475	157.96125	37.165	170.12875	58.670	204.65
1	37.0	237.150	158.1970	152.70100	27.871	164.52400	48.765	204.65
2	38.0	240.730	162.5650	149.18000	21.650	160.78000	42.150	204.65

Now that we have the new coordinates of each vehicle, we compute the relative distances of the ego-vehicle with other vehicles to verify if there is any relative distance that is less than 1 meter. Upon calculating the relative distances, it was observed that the distance between the ego vehicle and non-ego vehicle-1 was 0.848 meters, which is less than 1 meter, signifying a collision at the 37th second. The actual data reflects a relative distance of 0.925 meters at the 37th second, aligning with the collision time of 36.5 seconds in the simulation. This comparison between predicted and actual relative distances demonstrates the model's accuracy in predicting the collision at the precise second it occurred.

Relative distances of ego-vehicle with other vehicles on predicted data:

	rel_dist_non_ego_flow.1_x	rel_dist_non_ego_flow.2_x	rel_dist_non_ego_flow.3_x	rel_dist_non_ego_flow.4_x	rel_dist_non_ego_flow.5_x
0	15.467159	151.028249	126.446994	33.409504	98.693499
1	0.848008	156.460138	133.044807	34.229421	106.171825
2	14.729517	159.661025	137.155704	36.612876	111.255141

Relative distances of ego-vehicle with other vehicles on actual data:

	rel_dist_non_ego_flow.1_x	rel_dist_non_ego_flow.2_x	rel_dist_non_ego_flow.3_x	rel_dist_non_ego_flow.4_x	rel_dist_non_ego_flow.5_x
0	15.499313	150.933820	126.252338	33.192691	98.459601
1	0.925662	156.215342	132.260351	33.700930	105.022220
2	7.193652	161.265848	137.790775	35.719283	111.051574

For the timestamps of 36, 37, and 38 seconds, we determined the non-ego vehicles within a 50-meter radius of the ego vehicle, as illustrated below. It was found that at these specific three-second intervals, there are five vehicles located within a 50-meter proximity to the ego vehicle.

	36 sec	37 sec	38 sec
0	non ego flow.1	non ego flow.1	non ego flow.1
1	non ego flow.4	non ego flow.4	non ego flow.4
2	non ego flow.6	non ego flow.6	non ego flow.6
3	non ego flow.8	non ego flow.8	non ego flow.8
4	non ego flow.9	non ego flow.9	non ego flow.9

Conclusion: The Vector Error Correction Model (VECM) demonstrated strong performance in predicting the non-stationary time-series coordinates of vehicles. When these coordinates were used to calculate relative distances, the model precisely identified the collision occurrence step, matching the event in the simulation, especially when rounded to the nearest second. The next step will be trying the custom LSTM model with this type of dataset to compare statistical models with deep learning time-series models.